

I. REAL PARTY IN INTEREST	1
II. RELATED APPEALS AND INTERFERENCES.....	1
III. STATUS OF CLAIMS.....	2
IV. STATUS OF AMENDMENTS	2
V. SUMMARY OF CLAIMED SUBJECT MATTER.....	2
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.....	2
VII. ARGUMENT.....	3
VIII. CLAIMS APPENDIX.....	16
IX. EVIDENCE APPENDIX	21
X. RELATED PROCEEDINGS APPENDIX	22

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of : Customer Number: 46320

: Confirmation Number: 3998

Application No.: 10/612,613 : Group Art Unit: 2113

Filed: July 1, 2003 : Examiner: E. Mehrmanesh

For: CHECKPOINTING AND RESTARTING LONG RUNNING WEB SERVICES

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed December 20, 2006, wherein Appellants appeal from the Examiner's rejection of claims 1-24.

I. REAL PARTY IN INTEREST

This application is assigned to IBM Corporation by assignment recorded on July 1, 2003, at Reel 014258, Frame 0714.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1-24 are pending and finally rejected in this Application. It is from the final rejection of claims 1-24 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

The claims have not been amended subsequent to the imposition of the Second Office Action dated September 20, 2006 (hereinafter the Second Office Action).

V. SUMMARY OF CLAIMED SUBJECT MATTER

Referring to Figure 2 and to independent claim 1, a checkpoint processor 260 configured for coupling to individual Web services 240 through a Web services engine 250 is disclosed. The checkpoint processor 260 includes checkpoint logic 210, restart logic 220, and cleanup logic 230 (lines 5-6 of paragraph [0023] of Appellants' disclosure). The checkpoint logic 210 stores checkpoint data for the individual Web service instance invocations (lines 6-8 of paragraph [0023]). The restart logic 220 restores the stored checkpoint data to a replacement for failed ones of the individual Web service instance invocations (lines 9-11 of paragraph [0023]). The cleanup logic 230 removes the stored checkpoint data for concluded, non-failed ones of the individual Web service instance invocations (lines 11-13 of paragraph [0023]).

Referring to Figures 3A-3C and to independent claims 3 and 14, a method for managing checkpoints in a Web application is disclosed. In block 320, a state object for an invocation of a requesting Web service instance is stored (lines 5-11 of paragraph [0025]). In blocks 345/470, responsive to a failure in the Web service instance, a replacement Web service instance is

restarted and the state object is provided to a replacement Web service instance for the requesting Web service instance (lines 1-8 of paragraph [0028]; lines 1-9 of paragraph [0033]).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-2 were rejected under 35 U.S.C. § 101; and
2. Claims 1-24 were rejected under 35 U.S.C. § 102 for anticipation based upon Doyle et al., U.S. Patent Publication No. 2004/0243915.

VII. ARGUMENT

THE REJECTION OF CLAIMS 1 AND 2 UNDER THE 35 U.S.C. § 101

For convenience of the Honorable Board in addressing the rejections, claim 2 stands or falls together with independent claim 1.

Examiner's Initial Statement of the Rejection

In both the First Amendment dated April 3, 2006 (hereinafter the First Amendment) and in the Second Amendment, the Examiner asserted the following with regard to claims 1 and 2:

As per claims 1-2, claimed limitations of "*processor configured ...*" and "*logic programmed to...*" are not of statutory subject matter.

A processor configured or logic programmed to perform a method are merely software arrangements. The configuration of the processor is a computer program claimed as computer listings *per se*, i.e., the descriptions or expressions of the programs are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. (emphasis in original)

Appellants' Initial Arguments

In the Request for Reconsideration dated June 30, 2006 (hereinafter the Response), Appellants presented the following arguments with regard to the Examiner's rejection of claims 1

and 2 under 35 U.S.C. § 101 in the First Office Action. The Examiner's assertion that a "processor" does not constitute statutory matter is in direct contradiction to the plain language of 35 U.S.C. § 101. Claims 1 and 2 being directed to a machine is all that is required to satisfy the requirements of 35 U.S.C. § 101. In the decision of *In re Warmerdam*,¹ the Court concluded that the method claims recited in claims 1-4 "[involve] no more than the manipulation of basic mathematical constructs, the paradigmatic 'abstract idea.'" The Court then sustained the rejection of claims 1-4 under 35 U.S.C. § 101.

However, claim 5 of the same application recited "[a] machine having a memory which contains data representing a bubble hierarchy generated by the method of any of Claims 1 through 4." With regard to this claim, the Court stated that "[c]laim 5 is for a machine, and is clearly patentable subject matter" despite the prior finding that method being performed by the machine was non-statutory subject matter. Since claims 1 and 2 are directed to a processor (i.e., a machine), then claims 1 and 2 are directed to statutory subject matter within the meaning of 35 U.S.C. § 101.

Appellants also argued that utility of the claimed invention has been established. A discussion of the procedural considerations regarding a rejection based upon lack of utility (i.e., 35 U.S.C. § 101) is found in M.P.E.P. § 2107.02. Specifically, M.P.E.P. § 2107.02(I) states that:

regardless of the category of invention that is claimed (e.g., product or process), an applicant need only make one credible assertion of specific utility for the claimed invention to satisfy 35 U.S.C. 101 and 35 U.S.C. 112

¹ 33 F.3d 1354, 31 USPQ2d 1754 (Fed. Cir. 1994).

In the first paragraph of the "Detailed Description of the Preferred Embodiments" section in Appellants' disclosure, Appellants stated the following:

The present invention is a method and system for checkpointing and restarting long running Web services. In accordance with the present invention, individual Web service instances can checkpoint their respective states associated with particular invocations either periodically or in response to a triggering event. More particularly, the individual Web service instances can invoke checkpointing functionality included with or in association with an underlying Web services engine supporting each of the Web services instances. For each Web service instance invocation, a uniquely identifying instance identifier can be persisted in memory during the checkpointing process as can any state information forwarded by the Web service instance. Additionally a correlator able to identify an asynchronous communications session between the specific invocation and an invoking client further can be persisted in memory. Subsequently, when a restarting of a Web service instance invocation is required, for example in consequence of an application server crash, the restarted Web service instance invocation can be restored to its former state by uploading each of the instance identifier, the persisted state information and the asynchronous correlator.

Appellants, therefore, have asserted a credible utility (i.e., Web service instance invocation can be restored to its former state).

As noted in M.P.E.P. § 2107.02(III)(A), the Court of Customs and Patent Appeals in In re Langer² stated the following:

As a matter of Patent Office practice, a specification which contains a disclosure of utility which corresponds in scope to the subject matter sought to be patented must be taken as sufficient to satisfy the utility requirement of § 101 for the entire claimed subject matter unless there is a reason for one skilled in the art to question the objective truth of the statement of utility or its scope. (emphasis in original)

Since a credible utility is contained in Appellants' specification, the utility requirement of 35 U.S.C. § 101 (i.e., whether the invention produces a useful, concrete, and tangible result) has been met.

Examiner's Response

On pages 10 and 11 of the Second Office Action, in the section entitled "Response to Arguments," the Examiner responded to Appellants' arguments. Initially, the Examiner asserted

² 503 F.2d 1380, USPQ 288 (CCPA 1974).

the following:

With regards to the 35 USC § 101 rejection of claims 1-2, Examiner states that applicant's arguments are not persuasive and that the assertions with regards to the above claims are not legally in error as stated by the applicant. The plain meaning of claim language is directed to arrangement of software. Noting applicant's specifications (page 15, paragraph [0035]), which states "The present invention can be realized in hardware, software, or a combination of hardware and software." Also (specifications, page 8-9) and viewing figure 1, Web serves engine 150 and checkpoint processor 160 are interpreted as software and no hardware is disclosed. Applicant argues that claims 1-2 are directed to a machine is all that is required to satisfy the requirements of 35 USC § 101.

Notwithstanding the statement in Appellants' disclosure that "[t]he present invention can be realized in hardware, software, or a combination of hardware and software," claims 1 and 2 are not directed to software *per se*. In this regard, Appellants respectfully submit that the Examiner is espousing an outmoded view of the Patent Office that has been modified in response to the Honorable Board's decision in Ex parte Lundgren.³

In this regard, Appellants reference the latest revision of the M.P.E.P., which was issued in August of 2006. In particular, Appellants note that the M.P.E.P. §§ 2106, 2106.01, 2106.02 have been considerably rewritten. For example, whereas M.P.E.P. §§ 2106, 2106.01, 2106.02 were, in the previous revision of the M.P.E.P., entirely directed to statutory subject matter as it applies to computer-related inventions, only M.P.E.P. §§ 2106.01, 2106.02 are respectively directed to "Computer-Related Nonstatutory Subject Matter" and "Mathematical Algorithms" in the present revision. Moreover, M.P.E.P. § 2106 in the present revision is now directed to "Patent Subject Matter Eligibility," in general.

With regard to M.P.E.P. § 2106.01, entitled "Computer-Related Nonstatutory Subject Matter," Appellants note that the only discussion contained therein is with regard to functional

³ Appeal No. 2003-2088.

descriptive material and nonfunctional descriptive material. As stated therein:

Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*, 33 F.3d at 1360, 31 USPQ2d at 1759.

However, not only is the "processor" recited in the claims not descriptive material, the claims are not directed to descriptive material *per se*. Instead, a processor is a functional device and, thus, statutory subject matter under 35 U.S.C. § 101.

With regard to the requirements described in M.P.E.P. § 2106 for the Examiner to establish a *prima facie* case on the record, Appellants note that the Examiner has not determined whether the claimed invention falls with the judicial exceptions to 35 U.S.C. § 101. Moreover, the examiner has failed to "identify and explain in the record the reasons why a claim is for an abstract idea with no practical application." Thus, the Examiner has not established a proper rejection under 35 U.S.C. § 101.

THE REJECTION OF CLAIMS 1-24 UNDER 35 U.S.C. § 102 FOR ANTICIPATION BASED UPON DOYLE

For convenience of the Honorable Board in addressing the rejections, claim 2 stands or falls together with independent claim 1, and claims 4-24 stand or fall together with independent claim 3.

The factual determination of anticipation under 35 U.S.C. § 102 requires the identical disclosure, either explicitly or inherently, of each element of a claimed invention in a single reference.⁴ As part of this analysis, the Examiner must (a) identify the elements of the claims,

⁴ In re Rijekaert, 9 F.3d 1531, 28 USPQ2d 1955 (Fed. Cir. 1993); Lindermann Maschinenfabrik GMBH v. American

(b) determine the meaning of the elements in light of the specification and prosecution history, and (c) identify corresponding elements disclosed in the allegedly anticipating reference.⁵ This burden has not been met.

Claim 1

The Examiner asserted that the fail-over logic 170 of Doyle corresponds to the claimed checkpoint logic and that the optimization metrics 180, 230 corresponds to the claimed checkpoint data. The Examiner also asserted that the optimization logic 220 corresponds to the claimed restart logic. Appellants respectfully disagree.

At the outset, Appellants note that the Examiner has failed to establish that the fail-over logic of Doyle 170 "is programmed to store" the optimization metrics 180. Moreover, the optimization metrics are not used in a replacement for a failed individual Web service instance invocation. As noted in paragraphs [0039] and [0040], the optimization logic 220 identifies replacement nodes 240A, 240B for the failed service without using any of the information found in the optimization metrics 180, 230. In paragraph [0041], Doyle teaches that the service metrics 230 are used to place new instances of the services across one or more replacement nodes in an optimal fashion. However, this teaching fails to identically disclose the claimed "restore said stored checkpoint data to a replacement."

⁵ Hoist & Derrick Co., 730 F.2d 1452, 221 USPQ 481 (Fed. Cir. 1984).

⁵ Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick Co., *supra*.

Appellants originally presented the above arguments in the Response. The Examiner's response to these arguments are found in the first and second full paragraphs on page 11 of the Second Office Action and reproduced below:

As per claim 1 applicant argues that optimization metrics are not used in a replacement for a failed individual web service instance invocation. Examiner respectfully disagrees. Doyle discloses a method and system for managing autonomic failover in a services infrastructure, such as a Web services or grid services hosting infrastructure (page 2, paragraph [0019], lines 1-4). Noting paragraph [0036], line 1-4, grid coordinator (Fig. 1, element 150) can store optimization metrics (Fig. 1, element 180).

Doyle's optimization logic uses best-fit analysis applied to service metrics (page 4, paragraph [0045], lines 1-7). Paragraph [0036] describes optimization metrics can specify for each service instance various information such as revenue per unit of performance. The best-fit analysis uses service metrics and optimization metrics to determine replacement nodes. Figure 3 shows that upon identifying a replacement node, metrics are retrieved (Fig. 3, element 340). Paragraph [0042], lines 13-15 disclose using the service metrics 230 and platform metrics 250A, 250B to determine the replacement node.

For ease of reference, certain portions of claim 1 are reproduced below:

checkpoint logic programmed to store checkpoint data for the individual Web service instance invocations;

restart logic programmed to restore said stored checkpoint data to a replacement for failed ones of the individual Web service instance invocations.

Upon reviewing the Examiner's comments, Appellants note that the Examiner has still failed to addressed the issues raised by Appellants in the Response. Referring to the claimed language reproduced above, checkpoint data is stored and then the stored checkpoint data is restored to a replacement.

The Examiner's citation to paragraph [0019], lines 1-4 of Doyle is reproduced below:

The present invention is a method and system for managing autonomic failover in a services infrastructure, such as a Web services or grid services hosting infrastructure. In reference specifically to a grid services hosting infrastructure, a grid hosting node which has failed can be detected and a corresponding set of grid services hosted in the failed node can be determined.

As readily apparent, this passage is simply a very general description of Doyle that describes a "system for managing autonomic failover." However, this passage is silent as to the specific

claimed limitations reproduced above. Appellants do not disagree that the grid coordinator 150 can be used to store optimization metrics 180 associated with the operation of individual service instances 130A, 130B. However, paragraph [0036] is silent as to restoring checkpoint data to a replacement for failed ones of the individual Web service instance invocations, as claimed.

Regarding, the second paragraph of the Examiner's response, the Examiner's identification of "optimization logic" and citation of lines 1-7 of paragraph [0045] is silent as to claimed limitations at issue. For ease of reference, this passage is reproduced below:

A best-fit analysis can be applied to determine in which replacement nodes new instances of the services ought to be created. Notably, different elements of the service metrics can be weighted to emphasize certain elements over others, e.g. revenue per percentage of resource consumption is more important than preferred operating system.

Although Doyle teaches that "service metrics" may be used while determining "in which replacement nodes new instances of the services ought to be created," this passage is also silent as to restoring checkpoint data to a replacement for failed ones of the individual Web service instance invocations, as claimed. Using service metrics to determine a replacement node, as taught by Doyle, is not identical to restoring checkpoint data to a replacement, as claimed.

The Examiner's citation to Fig. 3 and assertion that "Figure 3 shows that upon identifying a replacement node, metrics are retrieved," also does not address the limitation at issue. Paragraph [0043] describes Fig. 3 and is reproduced below:

FIG. 3 is a flow chart illustrating a process for managing autonomic failover in the services grid of FIG. 1. Beginning in block 310, a node failure can be detected within the grid coordinator. In block 320, each service instance residing within the failed node can be identified. In block 330, one or more replacement nodes can be further identified. In block 340, the logged metrics for each of the service instances in the failed node can be retrieved. Alternatively, for each instance of a service hosted in the failed node, logged metrics for multiple instances across other nodes for the service can be retrieved so as to smooth anomalous events in any one instance for the service. In any case, platform metrics for each of the identified replacement nodes can be identified in block 350. (emphasis added)

Notwithstanding Doyle's teaching that the logged metrics for each of the service instances in a failed node are retrieved, Doyle fails to teach restoring checkpoint data to a replacement for failed ones of the individual Web service instance invocations, as claimed. Paragraphs [0044] and [0045] are also silent as to this limitation.

Finally, the Examiner cites lines 13-15 of paragraph [0042] and asserts that this passage "disclose using the service metrics 230 and platform metrics 250A, 250B to determine the replacement node." For ease of reference this passage is reproduced below:

To assist in the foregoing computation, a best-fit analysis can be applied to the service metrics 230 and platform metrics 250A, 250B.

This passage, although teaching using service metrics 230 and platform metrics 250A, 250B in the best-fit analysis, which is used to determine the replacement node in which the new instances of the services are created, is also silent as to restoring checkpoint data to a replacement for failed ones of the individual Web service instance invocations, as claimed. Using service metrics to determine a replacement node, as taught by Doyle, is not identical to restoring checkpoint data to a replacement, as claimed.

Therefore, for the reasons stated above, Appellants maintain their prior argument that Doyle fails to identically disclose the claimed "restore said stored checkpoint data to a replacement for failed ones of the individual Web service instance invocations."

With regard to the claimed "cleanup logic," the Examiner cited lines 16-20 of paragraph [0036] and lines 12-17 of paragraph [0045]. These passages cited by the Examiner, however, do

not identically disclose the claimed cleanup logic, which removes the stored checkpoint data "for concluded, non-failed ones of the individual Web service instance invocations." The passages cited by the Examiner are silent as to this particular limitation.

Appellants originally presented the above arguments in the Response. The Examiner's response to these arguments are found in the paragraph spanning pages 11 and 12 of the Second Office Action and reproduced below:

With regards to "cleanup logic" limitation of claim 1, Doyle discloses updating the optimization metrics with regards to individual service instances (page 3, paragraph [0036], lines 16-20). Updating the data in optimization metrics will save the last state of the individual service instances, which will clean up the prior data in the optimization metrics. (emphasis added)

At the outset, Appellants note that portion underlined in the Examiner's response is not explicitly taught by Doyle. For ease of reference both lines 16-20 of paragraph [0036] and lines 12-17 of paragraph [0045] are reproduced below:

Notably, the data included in the store of optimization metrics 180 can be updated regularly by operation of a monitor (not shown) coupled to the store of optimization metrics 180 which can collect performance data for the individual service instances 130A, 130B. (lines 16-20 of paragraph [0036])

Finally, the analysis of block 370 can be extended to pre-existing service instances within one or more of the replacement nodes where it is foreseeable that any one of the pre-existing service instances might become displaced through the placement of new services in the replacement nodes. (lines 12-17 of paragraph [0045])

The "cleanup logic" portion of claim 1 is reproduced below:

cleanup logic programmed to remove said stored checkpoint data for concluded, non-failed ones of the individual Web service instance invocations.

As evident from reviewing the cited passages within paragraphs [0036], [0045], Doyle fails to explicitly teach that "[u]pdating the data ... will clean up the prior data in the optimization," as asserted by the Examiner. Moreover, this passage is also completely silent as to teaching that the cleanup logic removes stored checkpoint data for concluded Web service

instance invocations. Therefore, the Examiner must be employing an inherency argument to establish that Doyle teaches these limitations.

However, such a reliance upon the doctrine of inherency to teach these missing features would be misplaced. Inherency may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient to establish inherency.⁶ To establish inherency, the extrinsic evidence must make clear that the missing element must necessarily be present in the thing described in the reference, and that the necessity of the feature's presence would be so recognized by persons of ordinary skill.⁷ The Examiner did not discharge that burden of indicating where such a teaching appears in the prior art. Thus, the Examiner has not established that these missing limitation are inherently disclosed by Doyle.

Therefore, for the reasons stated above, the Examiner has failed to establish that Doyle identically discloses the claimed invention, as recited in claim 1, within the meaning of 35 U.S.C. § 102.

Claims 3 and 14

Independent claims 3 and 14 each recite storing a state object for a Web service instance and restarting a replacement Web service instance while providing the state object to the

⁶ In re Rijckaert, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993) (reversed rejection because inherency was based on what would result due to optimization of conditions, not what was necessarily present in the prior art); In re Oelrich, 666 F.2d 578, 581-82, 212 USPQ 323, 326 (CCPA 1981).

⁷ Finnegan Corp. v. ITC, 180 F.3d 1354, 51 USPQ2d 1001 (Fed. Cir. 1999); In re Robertson, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999); Continental Can Co. USA v. Monsanto Co., 20 USPQ 2d 1746 (Fed. Cir. 1991); Ex parte Levy, 17 USPQ2d 1461 (BPAI 1990).

replacement Web service instance. In the First Office Action, the Examiner asserted that "individual platform metrics 250A, 250B" identically disclose the claimed Web service instance and that paragraph [0039] teaches that these individual platform metrics 250A, 250B are provided to a replacement Web service instance. Appellants respectfully disagree.

The individual platform metrics 240A, 250B are not described by Doyle as being for a requesting Web service instance that subsequently fails. Instead, Doyle teaches that the "[i]ndividual platform metrics 250A, 250B can be determined for each respective replacement node 240A, 240B." Thus, the individual platform metrics 240A, 250B are for the replacement node, not the failed node. Moreover, Doyle fails to teach that these individual platform metrics 240A, 250B are stored (i.e., "storing a state object ..."), as recited in the claims.

Appellants originally presented the above arguments in the Response. The Examiner's response to these arguments are found in the first full paragraph on page 12 of the Second Office Action and reproduced below:

As per claims 3 and 14, Examiner notes that platform metrics 250A, 250B are for replacement nodes, however Doyle discloses logging (e.g. "storing a state object") service metrics for each of the individual service instances (page 3, paragraph [0038], lines 6-9) and retrieving those logs for the failed one of the service instances (Fig. 3, element 340).

Similar to the arguments present above with regard to claim 1, Doyle does not teach "storing a state object" and "providing said state object to a replacement Web service instance" (i.e., comparable to the restoring checkpoint data to a replacement, as recited in claim 1). Instead, lines 6-9 of paragraph [0038] states that services instances are monitored and service metrics for these service instances are stored. However, absent from Doyle is a teaching that the service metrics are provided to a replacement Web service instance, as claimed. Thus, for the reasons

stated above, the Examiner has failed to establish that Doyle identically discloses the claimed invention, as recited in claims 3 and 14, within the meaning of 35 U.S.C. § 102.

Conclusion

Based upon the foregoing, Appellants respectfully submit that the Examiner's rejections under 35 U.S.C. §§ 101, 102 are not viable. Appellants, therefore, respectfully solicit the Honorable Board to reverse the Examiner's rejections under 35 U.S.C. §§ 101, 102.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due under 37 C.F.R. §§ 1.17, 41.20, and in connection with the filing of this paper, including extension of time fees, to Deposit Account 09-0461, and please credit any excess fees to such deposit account.

Date: February 16, 2007

Respectfully submitted,

/Scott D. Paul/
Scott D. Paul
Registration No. 42,984
Steven M. Greenberg
Registration No. 44,725
CUSTOMER NUMBER 46320

VIII. CLAIMS APPENDIX

1. A checkpoint processor configured for coupling to individual Web services through a Web services engine, said checkpoint processor comprising:

checkpoint logic programmed to store checkpoint data for the individual Web service instance invocations;

restart logic programmed to restore said stored checkpoint data to a replacement for failed ones of the individual Web service instance invocations; and,

cleanup logic programmed to removed said stored checkpoint data for concluded, non-failed ones of the individual Web service instance invocations.

2. The checkpoint processor of claim 1, further comprising logic for identifying an asynchronous correlator for each one of the individual Web service instance invocations and for storing said asynchronous correlator in association with corresponding ones of said stored checkpoint data.

3. A method for managing checkpoints in a Web application, the method comprising the steps of:

storing a state object for an invocation of a requesting Web service instance; and,

responsive to a failure in said Web service instance, restarting a replacement Web service instance and providing said state object to a replacement Web service instance for said requesting Web service instance.

4. The method of claim 3, wherein said storing step further comprises storing a unique identifier for said requesting Web service instance along with said stored state object.

5. The method of claim 4, wherein said storing step further comprises the steps of: identifying an asynchronous correlator for said invocation; and, storing said identified asynchronous correlator along with said stored state object.

6. The method of claim 3, wherein said storing step comprises the steps of: detecting a notable event in said Web service instance; and, responsive to said detection, storing a state object for an invocation of a requesting Web service instance.

7. The method of claim 3, wherein said storing step comprises the step of periodically storing a state object for an invocation of a requesting Web service instance.

8. The method of claim 4, wherein said step of providing further comprises the step of providing said unique identifier to said replacement Web service instance.

9. The method of claim 5, wherein said step of providing further comprises the step of providing said asynchronous correlator to said replacement Web service instance.

10. The method of claim 3, further comprising the step of discarding said stored state object when said Web service invocation has completed said invocation nominally.

11. The method of claim 3, further comprising the step of storing state data for a handler chain managing said Web service instance.

12. The method of claim 3, further comprising the steps of:
storing a residency indicator for said Web service instance invocation; and,
registering at least one selected event which when received causes an initiation of said restarting and providing steps.

13. The method of claim 3, wherein said step of restarting comprises the steps of:
determining whether an existing Web service instance can act as said replacement Web service instance; and,
if an existing Web service instance cannot be located which can act as said replacement Web service instance, instantiating a replacement Web service instance.

14. A machine readable storage having stored thereon a computer program for managing checkpoints in a Web application, the computer program comprising a routing set of instructions for causing the machine to perform the steps of:
storing a state object for an invocation of a requesting Web service instance; and,
responsive to a failure in said Web service instance, restarting a replacement Web service instance and providing said state object to a replacement Web service instance for said requesting Web service instance.

15. The machine readable storage of claim 14, wherein said storing step further comprises storing a unique identifier for said requesting Web service instance along with said stored state object.

16. The machine readable storage of claim 15, wherein said storing step further comprises the steps of:

identifying an asynchronous correlator for said invocation; and,

storing said identified asynchronous correlator along with said stored state object.

17. The machine readable storage of claim 14, wherein said storing step comprises the steps of:

detecting a notable event in said Web service instance; and,

responsive to said detection, storing a state object for an invocation of a requesting Web service instance.

18. The machine readable storage of claim 14, wherein said storing step comprises the step of periodically storing a state object for an invocation of a requesting Web service instance.

19. The machine readable storage of claim 15, wherein said step of providing further comprises the step of providing said unique identifier to said replacement Web service instance.

20. The machine readable storage of claim 16, wherein said step of providing further comprises the step of providing said asynchronous correlator to said replacement Web service instance.

21. The machine readable storage of claim 14, further comprising the step of discarding said stored state object when said Web service invocation has completed said invocation nominally.

22. The machine readable storage of claim 14, further comprising the step of storing state data for a handler chain managing said Web service instance.

23. The machine readable storage of claim 14, further comprising the steps of:
storing a residency indicator for said Web service instance invocation; and,
registering at least one selected event which when received causes an initiation of said restarting and providing steps.

24. The machine readable storage of claim 14, wherein said step of restarting comprises the steps of:

determining whether an existing Web service instance can act as said replacement Web service instance; and,

if an existing Web service instance cannot be located which can act as said replacement Web service instance, instantiating a replacement Web service instance.

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellants in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellants are unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.